DIAL: Distributed Interactive Analysis of Large datasets

BNL OMEGA talk

David Adams
BNL
August 5, 2002

Contents

- Definitions
- Use cases
- Requirements
- Design
- Datasets
- Dataset interface
- Dataset implementation
- Status and conclusions

Definitions

Dataset

- Collection of event data
  - Known event (beam crossing) ID's
  - Same content (raw, reconstructed, summary,…) for each event
  - Known luminosity and selection criteria (including triggers)
- Suitable for extracting physical quantities (cross section, limit, *etc*.)
  - Or special data for calibration, alignment or monitoring detector performance

Definitions (cont)

Large

- Too big to analyze from a single process
  - Today: 100 GB or more

Analysis

- Loop over events and perform the same action on each
  - Select events
  - Visualize events
  - Fill histograms and tuples
  - Generate new event data?

Definitions (cont)

Interactive

- Rapid response
  - Request processed in seconds, not hours
- Updates if the request is not finished quickly:
  - Partial results
  - Progress meter
    - % completed
    - Time to completion
  - Status visualization: what is being processed where
  - Able to terminate incomplete requests

Definitions (cont)

Distributed

- Central process presents results to the user
- Processing carried out by multiple jobs
- Jobs on different machines and different sites
- Motivation:
  - Access remote data
  - Parallel processing for faster response

Use cases

Event data specification

- User defines dataset
  - – which events and which data in each event
- Includes version of data for each event
  - – e.g. jets from reco version 14.2 instead of 13.1
- Restrict visible content of each event
  - – E.g. jets, not tracks
  - – Reduces cost of data access
- Dataset use as input for processing
- Dataset can be recorded and recalled later

Use cases (cont)

Event loop processing

- Event selection
  - User provides algorithm to be run on each event
  - Result determines if event is included in output dataset
- Fill histogram
  - User defines histogram and provides algorithm to fill from data for one event
- Fill tuple
  - Collection of named variables
  - User provides algorithm to fill 0-N times/event

Use cases (cont)

Single event processing

- Fetch event
  - – Data for selected event returned to user
  - – User may request a subset of the event data
- Visualization
  - – User defines a “view”
  - – User specifies an event and the associated data is used to fill the view

Use cases (cont)

Distributed processing

- Remote processing
  - – Analysis program run on the local node
  - – Data is located on a remote node
  - – Job processing data runs on the remote node
  - – User generates requests on the local node which are run on the remote node with results returned to the local node
- Parallel processing
  - – Dataset divided by event and each dataset is processed in a separate process or thread

Use cases (cont)

Distributed processing (cont)

- Multi-node processing
  - Previous processes are run on different compute nodes
- Multi-site processing
  - Previous processes are distributed over different sites
- GRID processing
  - Previous uses GRID for job specification, submission, authentication and monitoring

Requirements

Use cases

- Satisfy the preceding use cases

Interactivity

- Show status while a request is being processed
- Update status once/minute (adjustable)
- Return partial results on the same time scale
- Provide facility to abort a request

Requirements (cont)

History

- Event selection
  - Identify and record the attributes (including code) for each event selection algorithm
- Dataset
  - Identify and record each dataset
  - Provide mechanism to recover the selection algorithm(s) used to construct a dataset

Design

Dataset

- This description of a set of event data is the basis for all analysis
  - More on this later

Analyzer

- User works in an analysis framework which provides the tools required to view and process histograms and tuples
- ROOT is one example

Design (cont)

Task

- Specifies the operation to perform on each event including
  - Number of event selections to be performed
  - Histograms to be filled
  - Tuple to be filled
  - Code which makes selections and fills histograms and tuples

Design (cont)

Application

- Description of the executable run by jobs
- Loops over events in a dataset
- Executes task on each to generate event result
- Merges successful event results to form a dataset result
- Specification includes
  - Application name
    - E.g. Athena or ROOT
  - Version or acceptable versions

Design (cont)

Event result

- Flag indicating whether event was accepted for each event selection entry
- Histogram entries for each fill
- Tuple values for each fill
- Return status from task
  - – Success or failure

Design (cont)

Dataset result

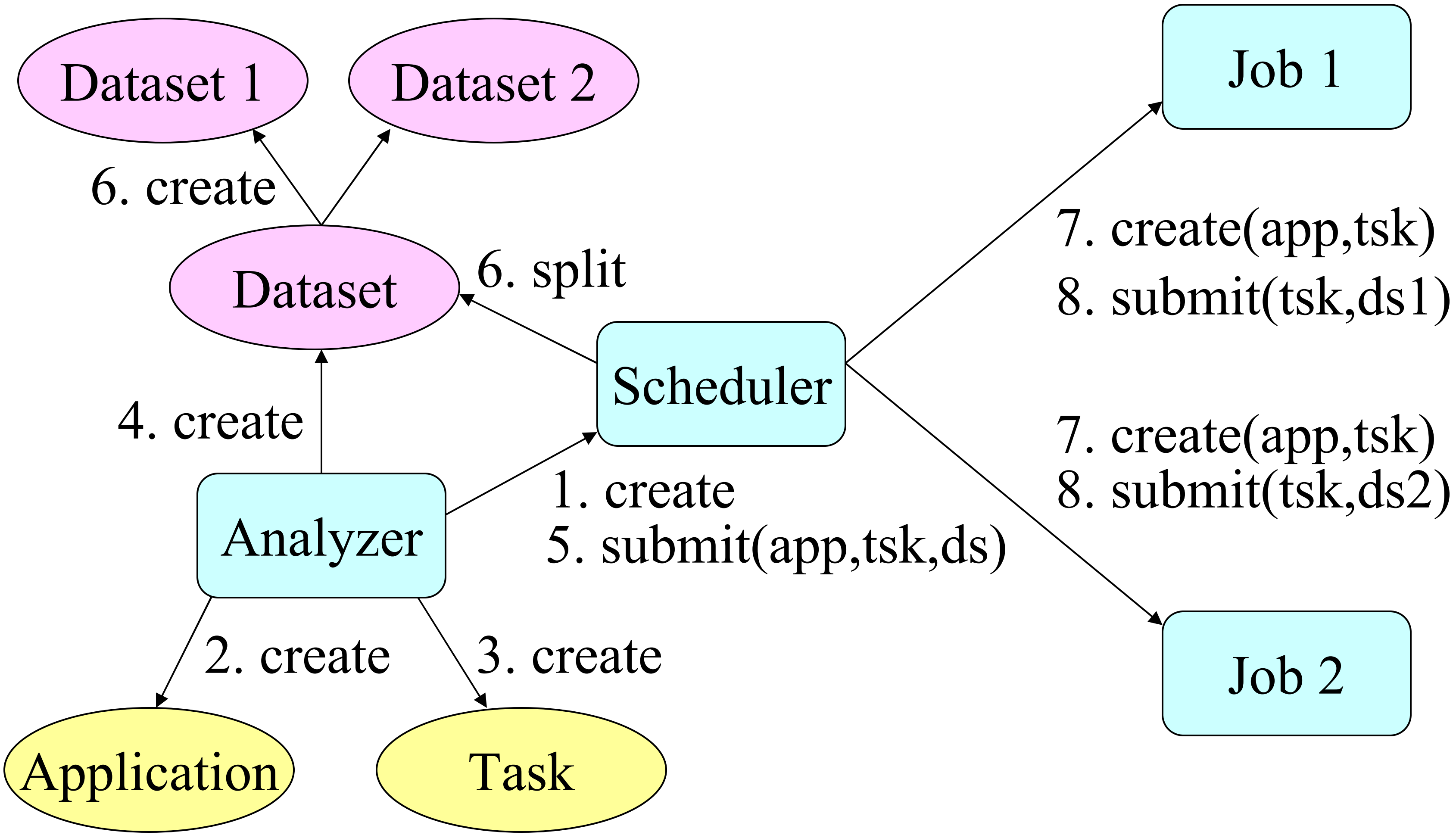
- New dataset for each event selection
  - Old dataset plus list of ID's for each event selection
- Filled histograms
- Filled tuples
- List of events for for which task processing was unsuccessful

Design (cont)

Job scheduler

- Receives request (application, task and dataset) from analyzer
- May divide dataset into sub-datasets
- Creates or locates jobs with a matching application (and possibly task)
- Adds task to jobs if needed
- Passes a dataset to each job, invokes task and receives result
- Merges results and returns to analyzer

Design (cont)

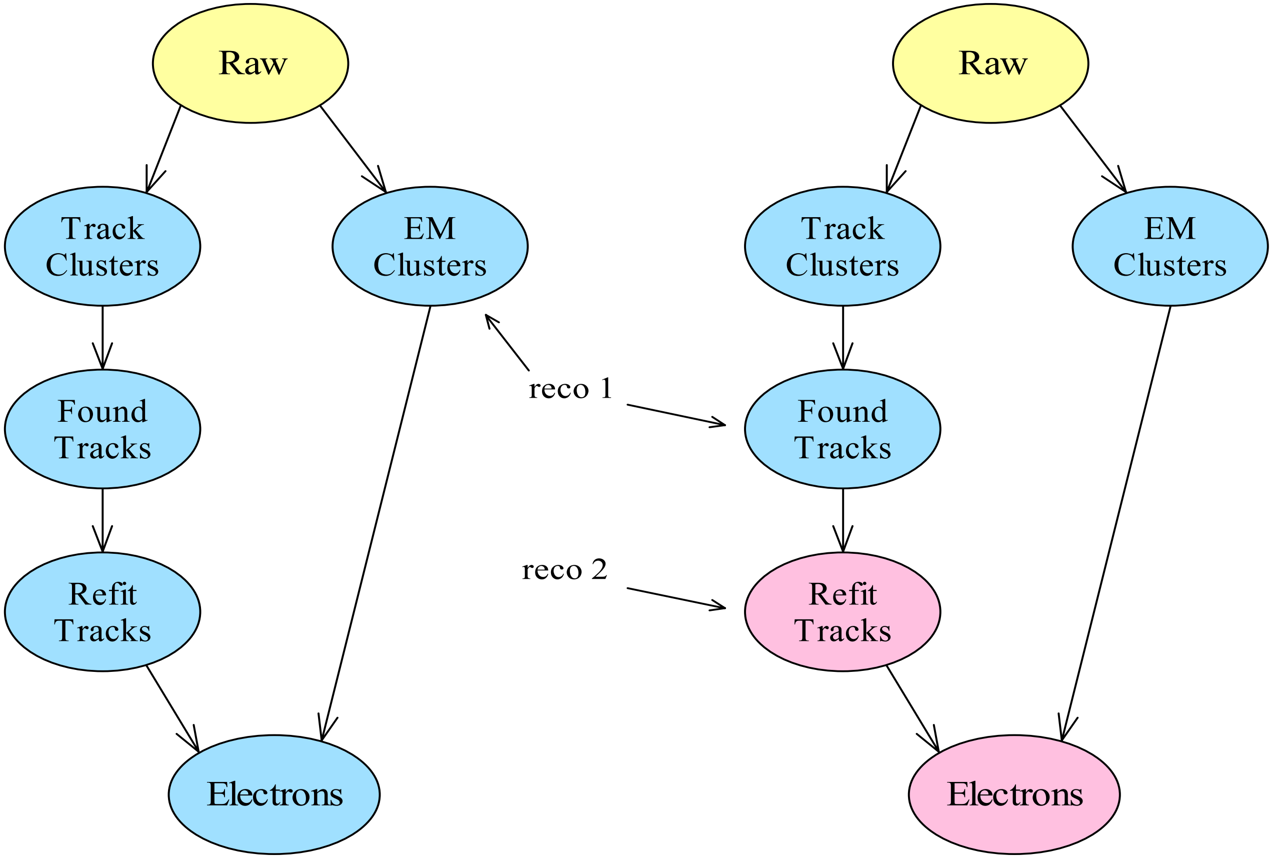

Dataset 1
Dataset 2
Job 1
6. create
Dataset
6. split
7. create(app,tsk)
8. submit(tsk,ds1)
Scheduler
4. create
7. create(app,tsk)
8. submit(tsk,ds2)
Analyzer
1. create
5. submit(app,tsk,ds)
2. create
3. create
Job 2
Application
Task

Datasets

Datasets provide interface and means for accessing event data

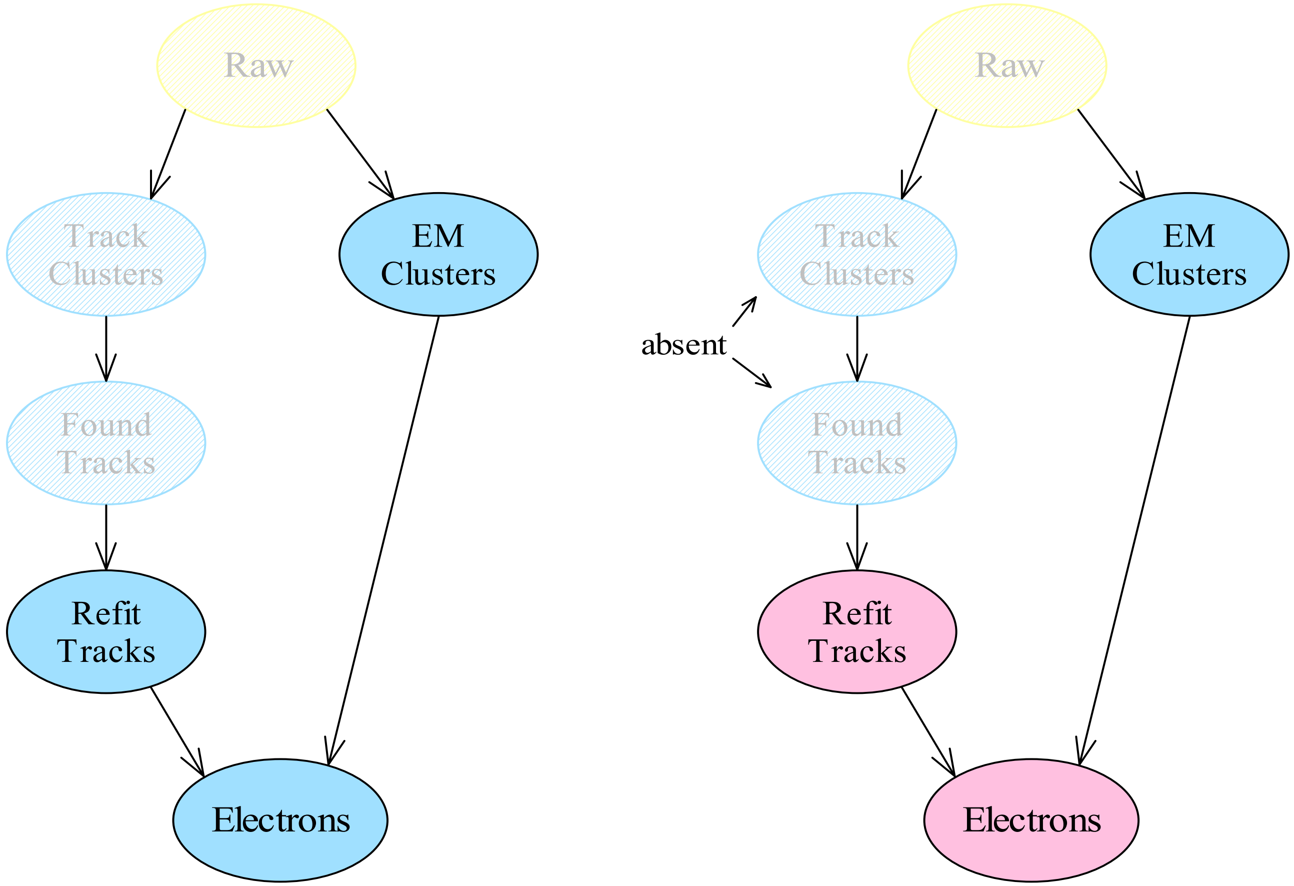
- Different types
  - Raw
  - Reconstructed
  - Summary
  - Tag
- Organized into EDO’s (event data objects)
  - Dataset does *not* see inside EDO
- Following plots give some examples

Datasets (cont)



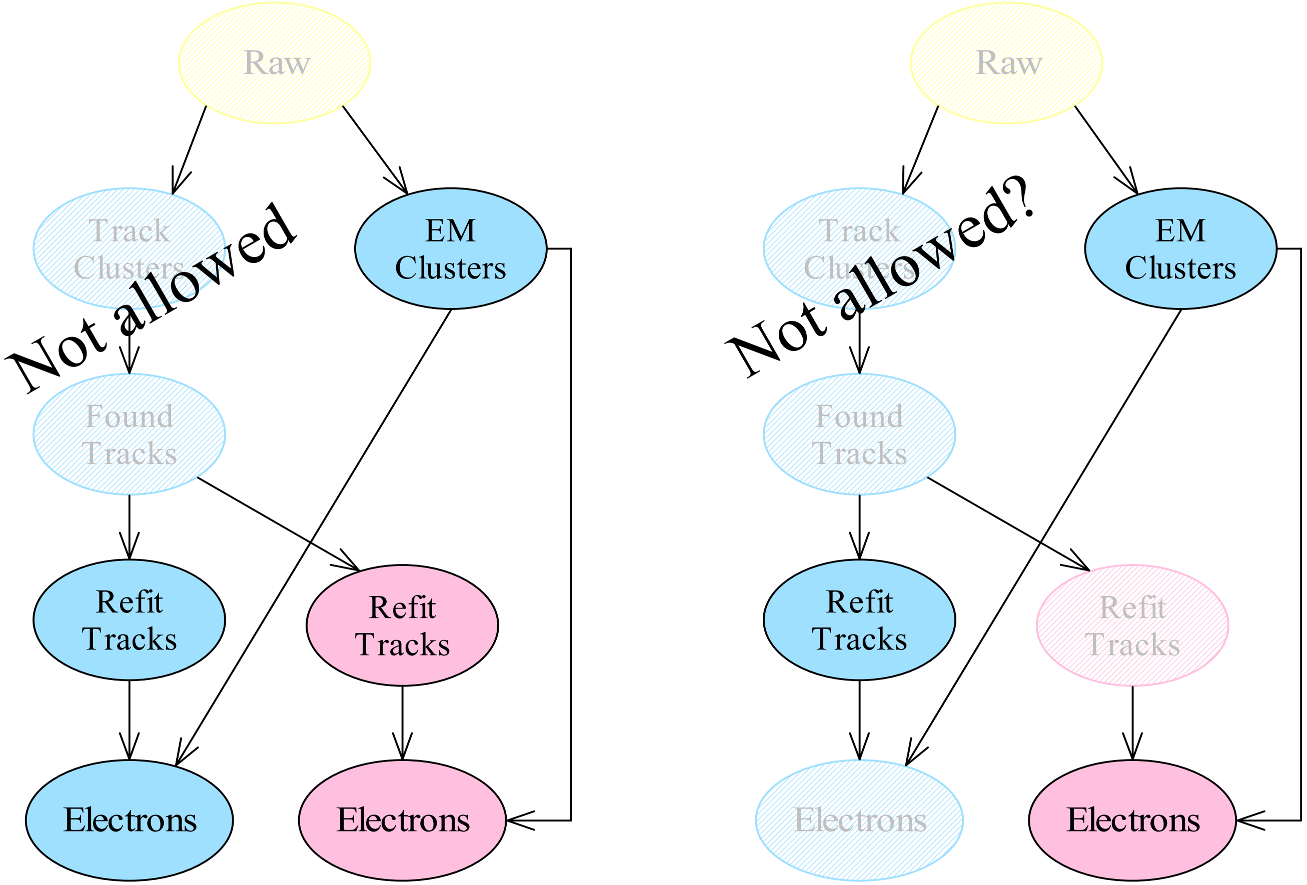
Two complete event views with the same content.

Datasets (cont)



Two incomplete and consistent event views with the same content.

Datasets (cont)



Ambiguous event view.

Inconsistent event view.

Dataset interface

Event range

- Collection of event ID’s

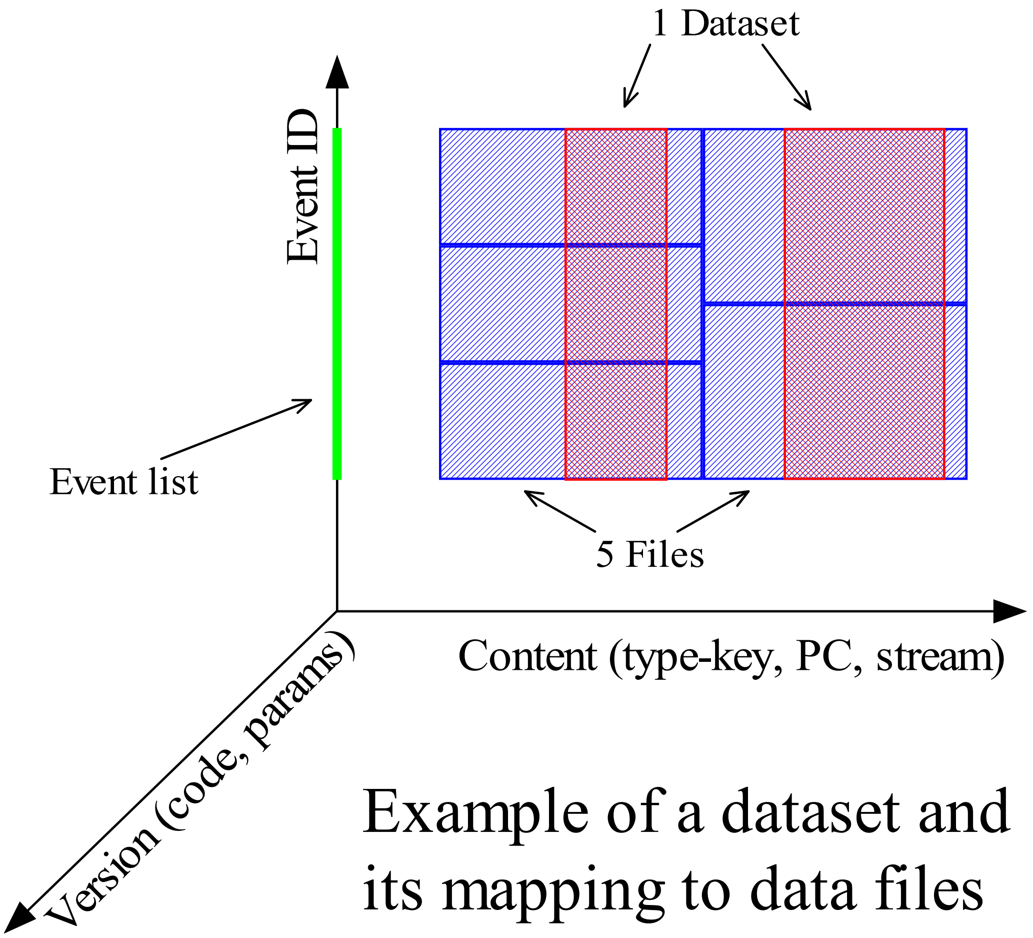
Content

- Collection of content ID’s

Event data (event views)

- For each event ID-content ID pair:
  - – A means to access the corresponding EDO or
  - – A flag indicating the EDO is not included
- No other event data is included

Dataset interface (cont)



Example of a dataset and its mapping to data files

Dataset implementation

Datasets are used in many ways

- Inspection by humans
- I/O for processing in C++
  - And other languages
- Cataloging in DB's

Implementation

- Prefer something object oriented
- At present, C++ classes with XML persistence

Status and conclusions

High-level design for DIAL is in place

- Described in this talk
- See http://www.usatlas.bnl.gov/~dladams/dial

Detailed design and first implementation of datasets is finished

- See http://www.usatlas.bnl.gov/~dladams/dataset